Web Browser Protection Profile

Version .5

April 20, 2001

Foreword

The base set of requirements used in this protection profile are taken from the "Common Criteria for Information Technology Security Evaluations, Version 2.1." Further information, including the status and updates of the Common Criteria can be found on the Internet at "http/cscr.nist.gov/cc/pp/pplist.html. Comments of the concerning this document should be directed to:

George Ryan, Pulse Engineering, Inc. (Team Lead)
Ella Miller, ACS Defense, Inc.
J.J. Rocha, Booz Allen, & Hamilton
Neal Ziring, NSA
Charles Lavine, The Aerospace Corporation

Acknowledgements

The authors would like to acknowledge The Aerospace Corporation for their preliminary work and reviewers of earlier drafts for their contributions.

Revision History

VERSION	EDITION	DATE	CHANGE PAGES	REASON
0.1	Draft	December 27, 1999	None	Initial Issue
0.2	Draft	January 6, 2000	None	Changes based upon comments
0.3	Draft	May 5, 2000	None	Changes based upon comments
0.4	Draft	January 29, 2001	None	Changes based upon comments
0.5	1st Submission	April 20, 2001	None	Changes based upon comments

Table of Contents

Conventions and Terminology

Conventions

Labels

The following label conventions are to aid in the referencing and understanding of assumptions, threats, policies, and objectives used throughout this document

Labeling Convention	Reference Category
A.<*name*>	Assumption
T.<*name*>	Threat
P.<*name*>	Organizational Security Policy (OSP)
O.<*name*>	Objective allocated to the TOE
OE.<*name*>	Objective allocated to the non-IT environment of the TOE
OIE.<*name*>	Objective allocated to the non-IT environment of TOE

Component Operations Conventions

The notation, formatting, and conventions used in this PP are largely consistent with those used in Version 2.1 of the Common Criteria (CC). Selected presentation choices are discussed here to aid the PP reader. The CC allows several operations to be performed on functional requirements: *assignment*, *selection*, *iteration* and *refinement* as defined in paragraph 2.1.4 of Part 2 of the CC. Each of these operations is used in this PP as discussed in the following sections.

Assignment and Selection

The assignment operation is used to assign a specific value to an unspecified parameter, such as the length of a password. The selection operation is used to select one or more options provided by the CC in stating a requirement. Completed assignment and selection operations are denoted by *italicised text*. Whenever an assignment or selection operation is left incomplete in this PP, that operation is denoted either by the text "*ST writer-provided assignment*" or by "*ST writer-provided selection*" respectively. These incomplete operations, along with their required parameters, appear in ***bold italicised*** *text.*

Iteration

Iteration of a component is required when an operation within the component must be completed multiple times with differing values, or for different allocation of functions to partitions of the TOE. Iterated functional and assurance components are given unique identifiers by appending a

semi-colon and an iteration number to the element identifiers from the CC. In cases where the text of each iteration would be identical, an asterisk (*) is given instead of an iteration number.

Refinement

The refinement operation is used to provide an elaboration of an existing CC element to explicitly meet stated objectives. Refinement of elements is denoted by bold text.

Application Notes

Application notes document guidance for how the requirement is expected to be applied. First, rather than being a separate section, the application notes have been integrated with requirements and indicated as notes. For each component, an application note may appear. For additional guidance, the CC itself should be consulted.

Requirements

In the requirement sections, each section that represents a requirement family or component, there is a mnemonic in parenthesis. These refer to the requirement section in the CC from which it was derived. Requirement elements have these references includes as superscripted text at the end of the element.

Terminology

Browser Administrator (referred to as TOE Administrator): A Host System User who is authorized to administer and configure the web browser. This may be the network or host platform administrator who is outside the scope of the TOE.

Browser Display Content: Content that is displayed in a browser window. It includes the text and images that are displayed and the underlying Hypertext Markup Language (HTML) page layout tags, but excludes executable content. Some display content may not be visible on the screen due to lack of contrast with backgrounds, overlaying of images, or clipping by the browser window.

Browser Mobile Code: Mobile code executed by a web browser or a web browser plug-in (e.g., Java, Javascript, VBScript, Portable Data File (PDF), ActiveX, Shockwave).

Browser User: A user of a web browser.

Certificate: A data structure that contains the necessary credentials to authenticate digital signatures and extract session keys from message headers, and that can be determined to be accurate through consultation with a trusted certifying agent. Integrity is usually ensured through the use of strong asymmetric encryption mechanisms.

Content: Information retrievable through a web server, or executed as a result of information requests to a web server. This includes information that can be requested via URLs, typically, but not limited to, HTML files, as well as Common Gateway Interface (CGI) scripts and server side includes.

Cookies: An http "cookie" is a small a small piece of information sent by a web server to be stored on a web browser so it can later be transmitted back from the browser. This is useful for having the

browser remember some specific information. An HTTP cookie cannot be used to get data from your hard drive, get your email or steal sensitive information about your person.

Digital Signature: A cryptographic hash of a data set (document, program, etc.) that has been encrypted with the signer's private key, such that having a trusted copy of the signer's public key enables the validation of the integrity of the data set, thus providing confidence that it has not been changed since it was signed. Certificates are used to guarantee the integrity of public keys from certification authorities, which vouch for the accurate assignment of public keys to signers.

Domain: A logical environment that allocates and separates resources for the simultaneous operation of the TOE and host platform.

Executable Content: Text or data embedded in or bound to documents or other datasets and executed without explicit end-user initiation.

Host System User: A user who is authorized to use the platform on which the web browser is executing.

HTTP: Hyper-Text Transfer Protocol, standardized as IETF RFC 2616.

HTTPS: HTTP transmitted over SSL or TLS.

Hyperlink: A Uniform Resources Layer (URL) embedded in a web page and displayed by a web browser so that a user of the browser can retrieve the page referenced by the URL by clicking on it with the mouse (or by some equivalent action).

Hypertext Document: A document containing hyperlinks, most commonly displayed by a web browser.

Mobile Code: The term given to software modules obtained from remote systems, transferred across a network, and then downloaded and executed on a local system, without explicit installation or initiation of execution by the recipient.

Plug-ins: A browser plug-in is a binary software module designed to add the ability to display or process new data types to a web browser.

SSL (Secure Sockets Layer): a cryptographic protocol that can provide integrity assurance, authentication, and confidentiality. There are two versions of SSL: version 2 and version 3; however, only version 3 will is considered for implementation by this protection profile.

TLS (**Transport Layer Security):** The successor to SSL, standardized as IETF RFC 2246.

URL (Uniform Resource Locator): Contains the host name and port number of the web server where a resource is located, and an access method and hierarchical specifier of a resource on that server. It is the syntax used to address resources on the WWW.

1 Introduction

This document is a Protection Profile for a web browser, conformant with the Common Criteria Version 2.1.

1.1 Identification

Title: Web Browser Protection Profile

Authors: U.S. Government and industry

Vetting Status:

CC Version: 2.1

General Status:

Registration:

Keywords: Browser, World Wide Web (WWW), Web, HTTP, HTTPS, Java, Javascript, ActiveX, Plug-in

1.2 Protection Profile Overview

This Protection Profile specifies the minimal security requirements for a web browser used in environments where access to and integrity of stored information (including the browser itself) residing on the servers as web page content, or transmitted between browser and server must be controlled and protected. The web browser must reliably determine the identity of the web server and its level of protection to the user. The browser must support mechanisms that provide non-repudiated identification of the browser user to the web server.

A conformant browser must isolate its users from interference by other users and protect the usage and content of browser data from unauthorized disclosure. It must control the execution of mobile code so that the host operating system is not corrupted, or user data incorrectly disclosed or corrupted. It must correctly apply the rules configured by the user or administrator to file uploads and downloads, cookie handling, secure session usage and marking, usage tracking (e.g., What's Related), user identification and authentication, execution of mobile code, and user isolation.

1.3 Document Organization

The following summarizes the contents of this protection profile:

Section 1 provides the introduction and establishes the context of this PP.

Section 2 defines the TOE description.

Section 3 contains the TOE Security Environment. This section defines the intended operating environment of the TOE through a related set of policies, threats and assumptions.

Section 4 provides the security objectives that support the assumptions and policies identified in the security environment.

Section 5 contains the functional and assurance requirements derived from the Common Criteria, Part 2 and 3, respectively, that support the security environment

Section 6 provides a rationale to explicitly demonstrate that the information technology security objectives satisfy the policies, assumptions and threats. This section explains how the sets of requirements satisfy the objectives, and that each security objective is addressed by one or more component requirements. Arguments are provided for the coverage of each objective.

Appendix A contains the list of acronyms within this PP.

Appendix B contains the list of reference documentation used to complete this

1.4 Related Protection Profiles

Controlled Access Protection Profile (CAPP): This PP provides the additional protection from network risks. The CAPP also specifies requirements for a product that enforces access controls on individual users and data objects.

Web Server Protection Profile: This PP provides the security requirements to support the web server and its interface component to the web browser.

2 TOE Description

The TOE serves as a PKI-enabled secure web browser compliant with the Global Information Grid (GIG) policy IA6-6510 and the Information Assurance Technical Framework (IATF) Forum documentation while communicating through the Hyper-Text Transfer Protocol - Secure (HTTPS).

The TOE is a hypertext document display application designed to retrieve data from a variety of types of web servers, and a graphical interface for accessibility to other web server services (e.g., File Transfer Protocol (FTP)). Through the implementation of PKI mechanisms, browser users will authenticate themselves to the web server resources they are attempting to access. The TOE has the capability to interface with other systems (e.g., Directory Services) for PKI certificate retrieval and storage. The TOE must only present theses certificate credentials to a web server upon initiation and for the duration of an established session, as defined by the browser user. Figure 2.1 provides an illustrative representation of the TOE's interface to web server.

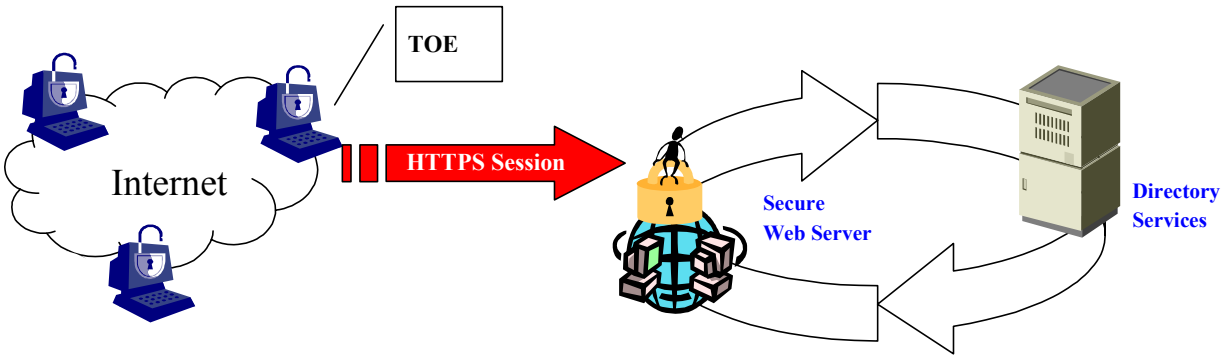

Figure 2.1. TOE Interface

The browser user will be required to provide evidence of authentication (e.g. PKI certificates) in order to access the web server resources and perform functions based upon the individual's specific role. Any error detected during the authentication process will prohibit the browser user from performing any further actions. The TOE will implement cryptographic protocols (i.e., Secure Sockets Layer (SSL) or Transport Layer Security (TLS)) so that information is transmitted and restricted from public access. These cryptographic protocols will allow the TOE and web server resources to exchange information in a secure manner. The secure exchange must support the encryption and decryption of data during its transfer between the server and TOE. This requires the secure negotiation of a session key, and the accurate indication to the browser user when encryption process is and is not invoked.

The browser user will be assigned various levels of access by the web server administrator in order to prevent unauthorized access to data, modification of data, or uploading of malicious code to the web server which could result in corruption of the web server resources and/or denial of service attacks for all browser users.

In addition to a graphical interface, the TOE may support capabilities that are generally hidden from the browser user. Some of these capabilities could be used to undermine the confidentiality of

user data, and subvert the user's control of access to host and user system resources. The TOE may support the execution of mobile code (e.g., JavaScript or ActiveX) in an isolated secure manner. Mobile code is downloaded from a web server as part of, or under direction of, a web page that executes within the TOE. This code will usually add visible content or interactive functionality to a web page. A conformant TOE must control the execution of mobile code by limiting its access to the underlying host system and user's data according to specifications set in the TOE configuration by ensuring that mobile code does not continue to execute once the session has been terminated.

A conformant TOE must also protect its configuration so mobile code cannot change the configuration of its security environment. A conformant TOE must also have the ability to disable mobile code, thereby preventing its unknown execution. As an additional security function of the TOE's confidentiality capability, it must be able to reliably disable the forwarding of cached web site history to third parties or vendors.

The TOE must support the isolation of multiple browser users from each other. This means that the content of web pages, the identity of web sites visited, and TOE configuration data must be stored separately for each browser user in a place that, by default, prevents access by other TOE or host users. Changes to the configuration of the TOE by one browser user should not affect the function of the TOE by other browser users. Likewise, web site user identities and credentials stored for one browser user must not be used in the authentication of another browser user. While supported by the underlying host platform for the storage and retrieval of downloaded or saved content (e.g., cache), the host platform is not included as part of the TOE. The underlying Host operating system, invoked applications, and/or communication mechanisms are outside of the secure protocol. In an effort to secure the data stored by the TOE, the browser user will rely on the control access mechanisms of the Host platform operating system.

The TOE may fully utilize "plug-ins" to support non-native data types (identified in the HTTP header with codes similar to Multipurpose Internet Mail Extensions (MIME) types). A conformant TOE must offer the browser user the choice of whether or not to install the plug-in, which has a security posture similar to that of other browser user-installed applications. Plug-ins are loaded dynamically from the local file system. Plug-ins register themselves during installation in a TOE's supported application table indicating which MIME-like types they support. Plug-ins can be downloaded from web sites and installed by the TOE. Plug-ins execute with the full authority of the browser user and without benefit of containment. Full trust is assigned by virtue of the plug-in's residence on the TOE's host platform.

3 TOE Security Environment

The TOE security environment will be determined by the security posture of the host platform operational environment. Local security organizations are responsible for adopting policies relevant to their respective operating environments, but no standards will be adopted at a level lower than U.S. Sensitive But Unclassified (SBU). Individuals assigned the responsibility of network security for the operational environment must ensure that protection safeguards are fully implemented on the environment on which the TOE will be utilized.

3.1 Secure Usage Assumptions

This section describes security aspects of the environment in which the TOE will be used or is intended to be used. This includes information about the physical, personnel, and connectivity aspects of the environment.

A.ADMIN_GUIDANCE
The TOE administrator will follow guidance regarding the configuration and maintenance of the TOE.

The TOE administrator is assumed to be a trustworthy and will follow all guidance documentation that outlines the proper configuration and scheduled maintenance of the TOE as stated by security policy. . Organization security policy will dictate who will serve in the capacity as the TOE administrator (e.g., user or network administrator).

A.AUTHORIZED_USE
Browser users will utilize information for intended purposes.

Information contained from the TOE will be used only for its authorized purpose(s).

A. NETWORK
Network connections used by browser users are secured from disclosure and unauthorized modification (e.g., by physical protection, encryption (ssh)).

Protocols exist that permit information to be protected from disclosure and modification during transmission. All configurations are sensitive and must be protected.

A.NON-MALICIOUS
Server executable content is non-malicious.

The assumption is that content providers will not intentionally install content that is known to be malicious. Although the TOE can limit, restrict, or audit the damage caused by hostile or defective code within the CGI scripts and other forms of server executable content, the TOE cannot prevent all compromises.

A.PLUG-IN
Browser-related programs known as plug-ins are not part of the TOE.

Plug-ins are not part of the TOE due to the fact that they are subprograms that are resident applications on the Host platform. Plug-ins are called by the TOE interface to enhance its functionality when viewing unrecognizable MIME-types.

A.ROLE_PERMISSION
Individual certificates working in conjunction with the TOE and accessed web servers, will determine the role permissions of the two roles: web users.

Roles are predefined based upon certificate attributes when the generated by the Certificate Management System (CMS).

A.TIME_SYNCH
The TOE will synchronize with the Host operating system in order to perform reliable timekeeping.

It is assumed that the TOE will rely on the operating system for the purpose of a timekeeping mechanism that will be used in conjunction with the CSP (e.g. certificate expiration validation).

3.2 Threats to Security

The threat possibilities addressed by the TOE are as follows:

T.ADMIN_ERROR
The browser administrator performs actions that result in an unauthorized browser user having access to browser information.

A browser administrator commits errors or fails to perform essential function and that may directly compromise organizational security objectives or changes the technical security policy enforced by the system or application.

T.EXPLOIT_ACTIVE_CONTENT
Web browser's active content performs operations on local platform that are undesired or unknown by browser user.

This is a broad category that includes most well known browser vulnerabilities related to the use of Java, Javascript, and ActiveX. The fundamental risk is a violation of user assumptions about the trust placed in active content and the ability of the browser to enforce expected constraints on the active content (e.g. a "sandbox" model, code signing, history-based access control, proff carrying code). Denials of service attacks are a special subset of this threat category that is common in well-known exploits (e.g., crashing the system, whiting out the screen).

T.EXPLOIT_USER_ACCESS
A hostile web application can escape the browser containment to access operating system files.

The security model for active content assumes that the browser provides isolation between operating system files and the TOE. This containment should preclude the TOE from accessing the platform system files. If an application is able to breach this containment, it can use the browser user's access rights to perform operations that are not permitted by the remote user.

T.HACKER_ACCESS

A hacker gains access due to the TOE allowing an unauthorized browser user to view browser information

The hacker may have the capability to view browser data by attacking the client during the authentication process, injecting viruses, or infecting the TOE with malicious code.

T.MALICIOUS_CODE
Authorized browser user or hacker stores or executes malicious code causing the TOE to function improperly.
The unauthorized and/or authorize user or attacker causes abnormal processing to occur that will violate the integrity, availability, or confidentiality of the TOE.

T.MOD_BROWSER_CONFIG
Unauthorized users modify browser security policy.

Potentially damaging code (e.g., viruses) could modify the files that reflect permissions for subsequent operations, or a rogue Java applet (if given permission erroneously) could modify policy files or security configuration preferences of an authorized user.

T.RESIDUAL_DATA
Browser user views residual data that had been written by a different browser user.

A browser user finds information that was left in the system by another browser user. Any time that a browser stores potentially sensitive information on disk (e.g., temporary files, cached application data, message buffers), it must be protected from access to other users. If private information is made available to other browser users, this can lead to unintended disclosure of user information. Once this flaw is discovered, it can be exploited to gather belonging to the system or another user.

T.SESSION_TERMINATE
An unattended active session of the TOE may results in an unauthorized user obtaining browser information or performing unauthorized functions.

The risk is that an authorized user could leave an established secured session unattended. This could allow an unauthorized user and/or attacker to access to browser information. Additionally, the unauthorized user may perform intentional or unintentional malicious acts under the identity of the browser user.

T.SECURE_DISPLAY
Deliberate or inadvertent misrepresentation of security-relevant system mode causes browser user to unknowingly violate security policy.

For example, TOE often display the security status of the current link (e.g., a closed padlock to indicate an SSL session). The knowledge of this status, or mode, tells the user if certain actions are permitted or not (e.g., sending passwords in the cleartext vs. encrypted transmissions) A display indicating SSL is active when in fact it is not could result in sensitive data being sent in the clear.

T.UNAUTH_ACCESS
Browser information (e.g., cookies, cache, history list) is read or modified by an unauthorized browser user.

This is essentially a violation of browser-related discretionary access control. Data associated with one user (e.g., cookie list or cache) is read or modified by an unauthorized user. This is particularly important for browsers used in public devices such as web kiosks or public telephones, or shared PCs (e.g., laptops, email facilities in airports, etc.).

3.3 Organizational Security Policies

An Organizational Security Policy is a set of rules or procedures imposed by an organization to protect its sensitive data. The set of security policies for the TOE is as follows:

P.ACCOUNTABILITY
TOE administrators and users shall be held accountable for actions that compromises the TOE.

The TOE administrators will be held accountable for defining policies, preferences, or properties that that may result in an unauthorized user viewing browser information. Likewise, TOE users shall be held accountable for their actions for accessing and/or modifying configuration preferences that may compromise the security posture of the TOE; particularly that which is shared by multiple users.

P.AVAILABILITY
Information shall be made available to authorized users as stated in the organizational security policies.

Information provided by the TOE and contained in the TOE will be available for authorized users through authenticated TOE sessions.

P.GUIDANCE
Guidance shall be provided for the secure installation and use of the TOE.

Guidelines will be established for installation and use of the TOE that will be provided to all browser users.

P.INFORMATION_AC
Information shall be accessed only by authorized individuals.

Only authorized browser users will be provided accessibility to the TOE based on access control mechanisms for stronger individual identification and authentication (e.g., certificates).

P.INTEGRITY
Browser information stored on the TOE shall retain its integrity by preventing modifications by unauthorized browser users.

This policy requires that the security access mechanisms shall be install to protect the browser information stored on the TOE.

P.PHYSICAL_CONTROL
Physical access to the host system for the TOE shall be controlled.

Although encryption is used to protect HTTP transmissions, it is out of the control of the TOE to require it for non-HTTP transmission. Thus, another objective must be physical controls, which prevent unauthorized access through the use of network sniffers, physical attacks, etc.

P.TRAIN
Authorized browser users shall be trained to utilize the TOE by implementing sound security policies and practices.

Authorized browser users must be trained to use the TOE properly.

4 Security Objectives

This section defines the security objectives of the TOE Security Functions (TSF) and it's supporting environment. Security objectives, categorized as either IT security objectives or non-IT security objectives, reflect the stated intent to counter identified threats and/or comply with any organizational security policies identified. All of the identified threats and organizational policies are addressed under one of the categories below.

4.1 Security Objectives for the TOE

O.ACCESS_CTRL
TOE administrators can determine the browser users permitted to access the browser data under their control.

The TOE administrators will limit the accessibility of the TOE to authorize web users, based on access control mechanisms for stronger individual identification and authentication (e.g., certificates).

O.ACTIVE_CONTENT
Active content downloaded from a web server is controlled.

This objective requires that the browser be capable of limiting the ability of active content (e.g. mobile code written in Java) to perform potentially harmful actions by limiting access to system and other user programs and data.

O.AUTOMATIC_FUNCTIONS
Browser recovers automatically to a consistent, secure state if a TSF does not complete successfully in the presence of certain types of failures (e.g., certificate validation failures or injection of malicious code).

The TOE will return to a static page display if failures of the TSF occur.

O.DATA_EXPOSURE
User information that is provided by the TOE to remote servers must be adequately protected from exposure.

Users can be deceived into violating security policy if they are not trained to understand the ways in which information can be disclosed when it is transmitted to a web server.

O.DATA_EXCH_CONF
The TSF shall enforce confidentiality of data exchanged between the web server and the TOE

The objective ensures that the data is protected by the use of encryption prior to being transmitted to/from the browser user.

O.I&A

The browser user shall be required to provide evidence (e.g., digital certificate) to be positively identified and authenticated to support accountability by the destination servers. This authentication, working in conjunction with server security mechanisms, shall provide for the capabilities of auditing each authenticated user.

This objective establishes the association of each transaction between an authenticated user and an application with a unique transaction ID. This allows events associated with a given transaction to be distinguished from other events involving the user and the application.

O.INTEGRITY
Browser users must be able to trust that the received information is from the expected originating source.
The goal of this objective is to provide browser users adequate confidence that the information received came from the expected source. Typically this is done by verification of the certificate held by the web server, which is transmitted as part of the HTTPS protocol.

O.ISOLATE
Browser user data (e.g., history, profiles, cookies, and cache) must be separate from other user data and not accessible by other users.

Many users may use the same workstation either simultaneously or at separate times to launch web browser sessions. User-specific data that is maintained for these user sessions must remain separate from other user sessions.

O.LABELS
Information shall be visually displayed, labeled or marked according to their content as received from the destination server.

The objective ensures that the TOE content will be displayed, labeled or marked in order to prevent the disclosure of data from secure web servers to unauthorized web users.

O.LIMIT_RESTRICTION
The TSF shall restrict the action of the web user prior to being authenticated.

The objective limits a browser user's actions (i.e., only a log on is allowed) prior to the TOE successfully communicating with a web server in order to verify the identity of the browser user.

O.SECURE_TRANS
Content shall be protected during transmission to and from web server.

Secure transmission protocols shall be employed by the TOE (i.e., SSL, TLS) in order to protect information from unauthorized disclosure or modification. While outside the scope of the PP, additional secure transmission countermeasures (e.g., VPNs) may be used by the host platform system and/or physical network components.

O.SECURITY_ROLES
The TSF shall maintain user privilege role separation.

This objective allows the TOE administrator to maintain security-relevant roles and the association of users with those roles.

O.SESSION_TERMINATION
The TOE shall terminate a session for inactivity during a secure established session.

The objective allows the TOE to terminate an inactive secure session after a specified interval of inactivity.

O.USER_ATTRIBUTES
The TOE shall maintain the user attributes for each active session.

The objective states that the TSF shall maintain security attributes associated with individual users in addition to browser user identity.

O.USER_AUTH_MGMT

The TSF shall manage user authorization privileges.

The TSF manages and updates user authorization and privilege data in accordance with organizational security and personnel policies.

O. VALIDATE
The TOE is capable of validating the origin source of the connected destination server.

In other words, the browser can, through the use of validation mechanisms (e.g., PKI server certificates) confirm the identity of the server source with which it is communicating.

4.2 IT Security Objective for the TOE Environment

OIE.CRYPTO_OPERATION
The TSF shall define the necessary settings to establish and maintain a security state of the TOE.
The objective defines the settings for each cryptographic operation.

OIE.CRYPTO_SERVICES
The TSF shall interface with the appropriate cryptographic services in order to validate the browser user authentication process.

The TOE must use various cryptographic services to authenticate the browser user to the server. These services include Certificate Authority (CA), certificates revocation processes, Directory Services, cryptographic tokens (e.g., smartcards, hardware tokens), and SSL services.

OIE.NETWORK_SERVICES
The TOE shall interface with other network devices to protect transmitted data.

This objective establishes the interfaces with various network devices (e.g., Directory Services, Firewalls) in order to protect the transmitted data to and from the TOE.

4.3 Non-IT Security Objective for the TOE Environment

O.GUIDANCE
Authorized administrators are given guidance and trained in the establishment and maintenance of sound security policies and practices.

Administrators must be trained to use and configure the system properly.

O.INSTALL
The TOE is delivered and installed in a manner that maintains system security.

A primary avenue of vulnerability is improperly configured systems. It is a required that the manufacturer of the TOE provides appropriate instructions for proper installation of the TOE. Additionally, organizational security policies shall dictate the standards of acceptable TOE configuration.

OE.MANAGE
The TOE shall be managed and operated in a manner that maintains system security.

The objective ensures that the TOE is maintained and updated on a consistent basis to prevent any malfunction that may compromise the information stored on the TOE.

OE.TOE_PROTECTION
The individual responsible for the TOE shall ensure the protection of computing resources and conflict resolution in a multitasking environment.

This objective ensures that the TOE is protected from overload in the case of performing simultaneous tasking with other host platform functions.

5 IT Security Requirements

Table 5.1 and the sections immediately following provides functional and assurance requirements that must be satisfied by a PP-compliant web browser. These requirements consist of functional components from Part 2 of the CC and an Evaluation Assurance Level (EAL) containing assurance components from Part 3.

5.1 Security Functional Requirements

Functional Class	Functional Components	
Cryptographic Support	FCS_COP.1	Cryptographic Operation
User Data Protection:	FDP_ACC.2	Complete Access Control
	FDP_ACF.1	Security Attribute Based Access Control
	FDP_DAU.2	Data Authentication with Identity of
	FDP_ETC.2	Export of User Data with Security
	FDP_IFC.2	Complete Information Flow Control
	FDP_IFF.1	Simple Security Attributes
	FDP_RIP.2	Full Residual Information Protection
	FDP_UCT.1	Basic Data Exchange Confidentiality
	FDP_UIT.1	Data Exchange Integrity
Identification and Authentication	FIA_AFL.1	Authenticated Failing Handling
	FIA_ATD.1	User Attribute Definition
	FIA_SOS.1	Specification of Secrets
	FIA_UAU.2	User Authentication Before Any Action
	FIA_UAU.3	Unforgeable Authentication
	FIA_UAU5	Multiple Authentication Mechanisms
	FIA_UID.2	User Identification Before Any Action
Security Management	FMT_MOF.1	Management of Security Functions Behavior (Enable/Disable)
	FMT_MSA.1	Management of Security Attributes
	FMT_MSA.3	Static Attribute Initialization
	FMT_REV.1	Revocation
	FMT_SAE.1	Time-Limited Authorization
	FMT.SMR.1	Security Management Roles

Table 5.1 Functional Requirements in the Web Browser Protection Profile

Functional Class	Functional Components	
Protection of the Trusted Security Functions	FPT_AMT.1	Abstract Machine Testing
	FPT_FLS.1	Fail Secure
	FPT_ITC.1	Inter-TSF Confidentiality During Transmission
	FPT_ITI.1	Inter-TSF Detection of Modification
	FPT_RCV.3	Trusted Recovery
	FPT_SEP.1	TSF Domain Separation
	FPT_STM.1	Reliable Time Stamps
	FPT_TRP.1	Trusted Path
	FPT_TST.1	TSF Testing
TOE Access	FTA_LSA.1	Limitation on Scope of Selectable Attributes
	FTA_SSL.3	TSF-Initiated Termination
	FTA_TSE.1	TOE Session Establishment

Table 5.1. (Cot'd) Functional Requirements in the Web Browser Protection Profile

5.1.1 Cryptographic Support (FCS)

5.1.1.1 Cryptographic Operation (FCS_COP.1)

The TSF shall perform *encryption/decryption transmission operations* in accordance with a specified cryptographic algorithm ***[ST writer-provided assignment: list the cryptographic algorithms]*** and *the following* cryptographic key size ***[ST writer-provided assignment: list the cryptographic key sizes]*** that meet the following *standards* ***of FIPS 140-1 level 3 and 4, as well as CAPP compliance.*** FCS_COP.1.1

Application Note:
The set of cryptographic operations is performed by the underlying security protocol implementation (e.g. SSL, TLS).

5.1.2 User Data Protection (FDP)

5.1.2.1 Complete Access Control (FDP_ACC.2)

The TSF shall enforce the *SFP* on *the individual user account information and profile settings* and all operations among subjects and objects covered by the SFP. FDP_ACC.2.1

The TSF shall ensure that all operations between any subject in the TSC and any object within the TSC are covered by an access control SFP. FDP_ACC.2.2

5.1.2.2 Security Attribute Based Access Control (FDP_ACF.1)

The TSF shall enforce the *access control SFP* to objects based on *user identity for the object's realm, TOE ACLs, and object access control information.* FDP_ACF.1.1

The TSF shall enforce the following rules to determine if an operation among controlled subjects and controlled objects is allowed: *Access control for an object shall be enforced for the attempted operation to permit only the authenticated web users to explicitly perform the operation.*[FDP_ACF.1.2]

5.1.2.3 Data Authentication with Identity of Guarantor (FDP_DAU.2)

The TSF shall provide a capability to generate evidence that can be used as a guarantee of the validity of the *user authentication*.[FDP_DAU.2.1]

The TSF shall provide *the web server* with the ability to verify evidence of the validity of the indicated information and the identity of the user that generated the evidence.[FDP_DAU.2.2]

5.1.2.4 Export of User Data with Security Attributes (FDP_ETC.2)

The TSF shall enforce the [assignment: *information flow control SFP(s*)] when exporting user data, controlled under the SFP(s), outside of the TSC.[FDP_ETC.2.1]

The TSF shall export the user data with the user data's associated security attributes**.**[FDP_ETC.2.2]

The TSF shall ensure that the security attributes, when exported outside the TSC, are unambiguously associated with the exported user data.[FDP_ETC.2.3]

The TSF shall enforce the following rules when user data is exported from the TSC: [assignment: additional exportation control rules].[FDP_ETC.2.4]

5.1.2.4 Complete Information Flow Control (FDP_IFC.2)

The TSF shall enforce and display *the sensitivity labels* on *applicable TOE content* (e.g., web pages accessed from destination web servers by authorized TOE users.)[FDP_IFC.2.1]

The TSF shall ensure that all operations that cause any information in the TSC to flow to and from any subject in the TSC are covered by an information flow control SFP.[FDP_IFC.2.2]

5.1.2.5 Simple Security Attributes (FDP_IFF.1)

The TSF shall enforce and display *sensitivity labels* based on *level of access from the requested server*.[FDP_IFF.1.1]

The TSF shall permit an information flow between a controlled subject and controlled information via a controlled operation based upon *the defined security level certificate attributes of the requested server.*[FDP_IFF.1.2]

The TSF shall enforce the *denial of access to web users that are not specifically defined in the destination server ACLs or do not have the required certificates attributes authorization*.[FDP_IFF.1.3]

The TSF shall provide *a visual notification disclaimer to the authorized web user that will require a keystroke acknowledgement sequence in order to access the controlled information*.[FDP_IFF.1.4]

The TSF shall explicitly authorize an information flow based on the *pre-defined certificate attributes that coincide with the destination server ACLs and authentication mechanisms established by the TOE administrator*.[FDP_IFF.1.5]

The TSF shall explicitly deny an information flow based on *the verification of web users to the aforementioned destination server ACLs, and directly provide a visual notification of such denial.*[*FDP*_IFF.1.6]

5.1.2.6 Full Residual Information Protection (FDP_RIP.2)

The TSF shall ensure that any previous information content of a resource is made unavailable upon the *deallocation of the resource fro*m all objects.[FDP_RIP.2.1]

5.1.2.7 Basic Data Exchange Confidentiality (FDP_UCT.1)

The TSF shall enforce the [assignment: *information flow control SFP(s*)] to be able to [selectio*n: transmit, receiv*e] objects in a manner protected from unauthorized disclosure.[FDP_UCT.1.1]

5.1.2.8 Data Exchange Integrity (FDP_UIT.1)

The TSF shall enforce the [assignment: *information flow control SFP(s*)] to be able to [selectio*n: transmit, receiv*e] user data in a manner protected from [selection: *modification*] errors.[FDP_UIT.1.1]

The TSF shall be able to determine on receipt of user data, whether [selection: *modificatio*n, *deletion, insertion, repla*y] has occurred. [FDP_UIT.1.2]

5.1.3 Identification and Authentication(FIA)

5.1.3.1 Authentication Failure Handling (FIA_AFL.1)

The TSF shall detect when *a TOE administrator defined number of* unsuccessful authentication attempts occur related to *established of authenticated sessions*.[FIA_AFL.1.1]

When the defined number of unsuccessful authentication attempts has been met or surpassed, the TSF shall *provide a visual notification of failure to the TOE user and prevent further access of the attempted server*.[FIA_AFL.1.2]

5.1.3.2 User Attribute Definition (FIA_ATD.1)

The TSF shall maintain the following list of security attributes belonging to individual users:

a. *TSF preferences specifying the authentication server (e.g., LDAP)*
b. *stored digital signature from the authenticated credentials*
c. ***(ST writer: add in any additional security attributes)***.[FIA_ATD.1.1]

5.1.3.3 Specification of Secrets (FIA_SOS.2)

The TSF shall provide a mechanism to generate secrets that meet : ***ST-defined quality metric***.[FIA_SOS.2.1]

The TSF shall be able to enforce the use of TSF generated secrets for *validation and authentication functions*].[FIA_SOS.2.2]

5.1.3.4 User Authentication Before Any Action (FIA_UAU.2)

The TSF shall require each user to be successfully authenticated before allowing any other TSF-mediated actions on behalf of that user.[FIA_UAU.2.1]

Application Note
The authentication of the user is required in a secure environment. If in an unsecure environment, the user will be able to perform all TSF-actions without being authenticated.

5.1.3.5 Unforgeable Authentication (FIA_UAU.3)

The TSF shall [selection: *detect, preven*t] use of authentication data that has been forged by any user of the TSF.[FIA_UAU.3.1]

The TSF shall [selection: *detect, preven*t] use of authentication data that has been copied from any other user of the TSF.[FIA_UAU.3.2]

5.1.3.6 Multiple Authentication Mechanisms (FIA_UAU.5)

The TSF shall provide *the requirement of client digital certificate and/or userid with password* to support user authentication.[FIA_UAU.5.1]
The TSF shall authenticate any user's claimed identity according to the *comparison of the stored digital certificate and user account information to that retained by the authentication server*.[FIA_UAU.5.2]

5.1.3.7 User Identification Before Any Action(FIA_UID.2)

The TSF shall require each user to identify itself before allowing any other TSF-mediated actions on behalf of that user.[FIA_UID.2.1]

5.1.4 Security Management(FMT)

5.1.4.1 Management of Security Functions Behavior (FMT_MOF.1)

The TSF shall restrict the ability to *modify the behavior of* the functions *of the individual user accounts* management to *the TOE administrators*.[FMT_MOF.1.1]

5.1.4.2 Secure Security Attributres (FMT_MSA.1)

The TSF shall enforce the *access control SFP* to restrict the ability to *modify* the security attributes of user accounts to the web browser administrator.[FMT_MSA.1.1]

5.1.4.3 Static Attributes Initialization(FMT_MSA.3)

The TSF shall enforce an access control SFP to provide restrictive default values for security attributes that are used to enforce the SFP.[FMT_MSA.3.1]

The TSF shall allow the TOE administrator to specify alternative initial values to override the default values when an object or information is created.[FMT_MSA.3.2]

5.1.4.4 Time-Limited Authorization (FMT_SAE.1)

The TSF shall restrict the capability to specify an expiration time for user authentication to web servers to the TOE administrator.[FMT_SAE.1.1]

For each of these security attributes, the TSF shall be able to terminate a web session and request re-authentication after the expiration time for the indicated security attribute has passed.[FMT_SAE.1.2]

5.1.4.5 Security Roles (FMT_SMR.1)

The TSF shall maintain the role *of TOE administrator*.[FMT_SMR.1.1]

The TSF shall be able to associate users with roles.[FMT_SMR.1.2]

5.1.5 Protection of the TSF (FPT)

5.1.5.1 Abstract Machine Testing (FPT_AMT.1)

The TSF shall run a suite of tests during initial start-up and periodically during normal operation to demonstrate the correct operation of the security assumptions provided by the abstract machine that underlies the TSF.[FPT_AMT.1.1]

5.1.5.2 Fail with Preservation of Secure State (FPT_FLS.1)

The TSF shall preserve a secure state when the following types of failures occur:

a) session interruption
b) hardware failure
c) operational environment failure (e.g., power outage).[FPT_FLS.1.1]

5.1.5.3 Inter-TSF Confidentiality During Transmission (FPT_ITC.1)

The TSF shall protect all TSF data transmitted from the TSF to a remote trusted IT product from unauthorized disclosure during transmission. [FPT_ITC.1.1]

5.1.5.4 Inter-TSF Detection of Modification (FPT_ITI.1)

The TSF shall provide the capability to detect modification of all TSF data during transmission between the TSF and a remote trusted IT product within the following metric: *an algorithmic checksum comparison*.[FPT_ITI.1.1]

The TSF shall provide the capability to verify the integrity of all TSF data transmitted between the TSF and a remote trusted IT product and perform *rejection of data and notification to the TOE user* if modifications are detected.[FPT_ITI.1.2]

5.1.5.5 Automated Recovery without Undue Loss (FPT_RCV.3)

When automated recovery from a failure or service discontinuity is not possible, the TOE shall enter a maintenance mode where the ability to return the TOE to a secure state is provided.[FPT_RCV.3.1]

For *web session failures*, the TOE shall ensure the return of the TOE to a secure state using automated procedures.[FPT_RCV.3.2]

The functions provided by the TSF to recover from failure or service discontinuity shall ensure that the secure initial state is restored without exceeding a *predefined time interval* by the TOE administrator for loss of TSF data or objects within the TSC.[FPT_RCV.3.3]

The TSF shall provide the capability to determine the objects that were or were not capable of being recovered.[FPT_RCV.3.4]

5.1.5.6 TSF Domain Separation (FPT_SEP.1)

The TSF shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.[FPT_SEP.1.1]

The TSF shall enforce separation between the security domains of subjects in the TSC.[FPT_SEP.1.2]

5.1.5.5 Reliable Time Stamps (FPT_STM.1)

The TSF shall be able to provide reliable time stamps for its own use.[FPT_STM.1]

5.1.5.6 TSF Testing (FPT_TST.1)

The TSF shall run a suite of self-tests during initial start-up and periodically during normal operation to demonstrate the correct operation of the TSF.[FPT_TST.1.1]

The TSF shall provide authorized users with the capability to verify the integrity of the TSF data.[FPT_TST.1.2]

The TSF shall provide authorized users with the capability to verify the integrity of stored TSF executable code.[FPT_TST.1.3]

5.1.6 TOE Access (FTA)

5.1.6.1 Limitation on Scope of Selectable Attributes (FTA_LSA.1)

The TSF shall restrict the scope of the session security attributes *authentication credentials* based on *the individual assignment and/or role as determined by the administrator*.[FTA_LSA.1.1]

5.1.6.2 TSF-Initiated Termination (FTA_SSL.3)

The TSF shall terminate an interactive session after a *time interval of user inactivity pre-defined by the TOE administrator*.[FTA_SSL.3.1]

5.1.6.3 TOE Session Establishment (FTA_TSE.1)

The TSF shall be able to deny session establishment based on *the established tables of the Access Control List (ACL) and individual access*.[FTA_TSE.1.1]

5.1.6.4 Trusted Path (FTP_TRP.1)

The TSF shall provide a communication path between itself and destination web servers that is logically distinct from other communication paths and provides assured identification of its end points and protection of the communicated data from modification or disclosure. [FTP_TRP.1.1]

The TSF shall permit *the TSF* to initiate communication via the trusted path. FTP_TRP.1.2

The TSF shall require the use of the trusted path for *initial user authentication, and cryptologic services*. FTP_TRP.1.3

5.2 Security Assurance Requirements

This section defines the assurance requirements for the TOE. Assurance requirement components are EAL 2, with no augmentation, from Part 3 of the CC. These requirements are summarized in Table 5.2.

Assurance Class	Assurance Components	
Configuration Management (ACM)	ACM_CAP.2	Configuration Items
Delivery and Operation (ADO)	ADO_DEL.1	Delivery Procedures
	ADO_IGS.1	Installation, Generation, and Start-up Procedures
Development (ADV)	ADV_FSP.1	Informal Functional Specification
	ADV_HLD.1	Descriptive High-Level Design
	ADV_RCR.1	Informal Correspondence Documentation
Guidance Documents (AGD)	AGD_ADM.1	Administrator Guidance
	AGD_USR.1	User Guidance
Tests (ATE)	ATE_COV.1	Evidence of Coverage
	ATE_FUN.1	Functional Testing
	ATE_IND.2	Independent Testing - Sample
AVA	AVA_SOF.1	Strength of TOE Security Function Evaluation
	AVA_VLA.1	Developer Vulnerability Analysis

Table 5.2. Assurance Requirements in the Web Browser Protection Profile

5.2.1 Configuration Management (ACM)

5.2.1.1 Configuration Items (ACM_CAP.2)

The reference for the TOE shall be unique to each version of the TOE. ACM_CAP.2.1C

The developer shall provide a reference for the TOE. ACM_CAP.2.1D

The TOE shall be labeled with its reference. ACM_CAP.2.2C

The developer shall use a CM system. ACM_CAP.2.2D

The CM documentation shall include a configuration list. ACM_CAP.2.3C

The developer shall provide CM documentation. ACM_CAP.2.3D

The configuration list shall describe the configuration items that comprise the TOE.[ACM_CAP.2.4C]

The CM documentation shall describe the method used to uniquely identify the configuration items.[ACM_CAP.2.5C]

The CM system shall uniquely identify all configuration items.[ACM_CAP.2.6C]

5.2.2 Delivery and Operation (ADO)

5.2.2.1 Delivery Procedures (ADO_DEL.1)

The delivery documentation shall describe all procedures that are necessary to maintain security when distributing versions of the TOE to a user's site.[ADO_DEL.1.1C]

The developer shall document procedures for delivery of the TOE or parts of it to the user.[ADO_DEL.1.1D]

The developer shall use the delivery procedures.[ADO_DEL.1.2D]

5.2.2.2 Installation, Generation, and Start-Up Procedures (ADO_IGS.1)

The documentation shall describe the steps necessary for secure installation, generation, and start-up of the TOE.[ADO_IGS.1.1C]

The developer shall document procedures necessary for the secure installation, generation, and start-up of the TOE.[ADO_IGS.1.1D]

5.2.3 Development (ADV)

5.2.3.1 Informal Functional Specification (ADV_FSP.1)

The functional specification shall describe the TSF and its external interfaces using an informal style.[ADV_FSP.1.1C]

The developer shall provide a functional specification.[ADV_FSP.1.1D]

The functional specification shall be internally consistent.[ADV_FSP.1.2C]

The functional specification shall describe the purpose and method of use of all external TSF interfaces, providing details of effects, exceptions and error messages, as appropriate.[ADV_FSP.1.3C]

The functional specification shall completely represent the TSF.[ADV_FSP.1.4C]

5.2.3.2 Descriptive High-Level Design (ADV_HLD.1)

The presentation of the high-level design shall be informal.[ADV_HLD.1.1C]

The developer shall provide the high-level design of the TSF.[ADV_HLD.1.1D]

The high-level design shall be internally consistent.[ADV_HLD.1.2C]

The high-level design shall describe the structure of the TSF in terms of subsystems.[ADV_HLD.1.3C]

The high-level design shall describe the security functionality provided by each subsystem of the TSF.[ADV_HLD.1.4C]

The high-level design shall identify any underlying hardware, firmware, and/or software required by the TSF with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.[ADV_HLD.1.5C]

The high-level design shall identify all interfaces to the subsystems of the TSF.[ADV_HLD.1.6C]

The high-level design shall identify which of the interfaces to the subsystems of the TSF are externally visible.[ADV_HLD.1.7C]

5.2.3.3 Informal Correspondence Demonstration (ADV_RCR.1)

For each adjacent pair of provided TSF representations, the analysis shall demonstrate that all relevant security functionality of the more abstract TSF representation is correctly and completely refined in the less abstract TSF representation.[ADV_RCR.1.1C]

The developer shall provide an analysis of correspondence between all adjacent pairs of TSF representations that are provided.[ADV_RCR.1.1D]

5.2.4 Guidance Documents (AGD)

5.2.4.1 Administrator Guidance (AGD_ADM.1)

The administrator guidance shall describe the administrative functions and interfaces available to the administrator of the TOE.[AGD_ADM.1.1C]

The developer shall provide administrator guidance addressed to system administrative personnel.[AGD_ADM.1.1D]

The administrator guidance shall describe how to administer the TOE in a secure manner.[AGD_ADM.1.2C]

The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.[AGD_ADM.1.3C]

The administrator guidance shall describe all assumptions regarding user behaviour that are relevant to secure operation of the TOE.[AGD_ADM.1.4C]

The administrator guidance shall describe all security parameters under the control of the administrator, indicating secure values as appropriate.[AGD_ADM.1.5C]

The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the TSF.[AGD_ADM.1.6C]

The administrator guidance shall be consistent with all other documentation supplied for evaluation.[AGD_ADM.1.7C]

The administrator guidance shall describe all security requirements for the IT environment that are relevant to the administrator.[AGD_ADM.1.8C]

5.2.4.2 User Guidance (AGD_USR.1)

The user guidance shall describe the functions and interfaces available to the non-administrative users of the TOE.[AGD_USR.1.1C]

The developer shall provide user guidance.[AGD_USR.1.1D]

The user guidance shall describe the use of user-accessible security functions provided by the TOE.[AGD_USR.1.2C]

The user guidance shall contain warnings about user-accessible functions and privileges that should be controlled in a secure processing environment.[AGD_USR.1.3C]

The user guidance shall clearly present all user responsibilities necessary for secure operation of the TOE, including those related to assumptions regarding user behaviour found in the statement of TOE security environment.[AGD_USR.1.4C]

The user guidance shall be consistent with all other documentation supplied for evaluation.[AGD_USR.1.5C]

The user guidance shall describe all security requirements for the IT environment that are relevant to the user.[AGD_USR.1.6C]

5.2.5 Tests (ATE)

5.2.5.1 Evidence of Coverage (ATE_COV.1)

The evidence of the test coverage shall show the correspondence between the tests identified in the test documentation and the TSF as described in the functional specification.[ATE_COV.1.1C]

The developer shall provide evidence of the test coverage.[ATE_COV.1.1D]

5.2.5.2 Functional Testing (ATE_FUN.1)

The test documentation shall consist of test plans, test procedure descriptions, expected test results and actual test results.[ATE_FUN.1.1C]

The developer shall test the TSF and document the results.[ATE_FUN.1.1D]

The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.[ATE_FUN.1.2C]

The developer shall provide test documentation.[ATE_FUN.1.2D]

The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. These scenarios shall include any ordering dependencies on the results of other tests.[ATE_FUN.1.3C]

The expected test results shall show the anticipated outputs from a successful execution of the tests.[ATE_FUN.1.4C]

The test results from the developer execution of the tests shall demonstrate that each tested security function behaved as specified.[ATE_FUN.1.5C]

5.2.5.3 Independent Testing - Sample (ATE_IND.2)

The TOE shall be suitable for testing[ATE_IND.2.1C]

The developer shall provide the TOE for testing.[ATE_IND.2.1D]

The developer shall provide an equivalent set of resources to those that were used in the developer's functional testing of the TSF.[ATE_IND.2.2C]

5.2.6 Vulnerability Assessment (AVA)

5.2.6.1 Strength of TOE Security Function Evaluation (AVA_SOF.1)

For each mechanism with a strength of TOE security function claim the strength of TOE security function analysis shall show that it meets or exceeds the minimum strength level defined in the PP/ST.[AVA_SOF.1.1C]

The developer shall perform a strength of TOE security function analysis for each mechanism identified in the ST as having a strength of TOE security function claim.[AVA_SOF.1.1D]

Application Note:
Strength of Function Claims for this Protection Profile are specified Table 5.3

Mechanism	**Minimum Strength Level**	**Strength of Function Metric**
Password used for access control under the WEBUSER SFP	SOF-Medium	None
Certificates	SOF-Medium	None

Table 5.3. Strength of Function Claims for the Web Browser Protection Profile

5.2.6.2 Developer Vulnerability Analysis (AVA_VLA.1)

The documentation shall show, for all identified vulnerabilities, that the vulnerability cannot be exploited in the intended environment for the TOE.[AVA_VLA.1.1C]

The developer shall perform and document an analysis of the TOE deliverables searching for obvious ways in which a user can violate the TSP.[AVA_VLA.1.1D]

The developer shall document the disposition of obvious vulnerabilities.[AVA_VLA.1.2D]

6 Rationale

This section provides the rationale for the selection, creation, and use of the security policies, objectives, and functional requirements.

6.1 Security Objectives Rationale

This section provides a rationale for the existence of each threat, policy statement, security objective, and functional requirement that is comprised within this protection profile. Table 6.1 shows the objective to policy/threat/assumption mapping.

Threats/Policies	Objectives
T.ADMIN_ERROR	O.I&A; O_ISOLATE; O.SECURITY_ROLES
T.EXPLOIT_ACTIVE_CONTENT	O.ACCESS_CTRL; O.ACTIVE_CONTENT
T.EXPLOIT_USER_ACCESS	O.ACCESS_CTRL; OIE.CRYPTO_SERVICES
T.HACKER_ACCESS	O.DATA_EXCH_CONF; O.DATA_EXPOSURE; O_I&A; O.INTEGRITY; O.SESSION_TERMINATION; OIE.CRYPTO_SERVICES;
T.MALICIOUS_CODE	ACCESS_CNTRL; O.AUTOMATIC_FUNCTIONS; O.I&A; O.INTEGRITY
T.MOD_BROWSER_CONFIG	O.ACCESS_CTRL; O.I&A; O.USER_AUTH_MGMT
T.RESIDUAL_DATA	O.ISOLATE
T.SESSION_TERMINATE	O.AUTOMATIC_FUNCTIONS; O.SESSION_TERMINATION ;
T.SECURE_DISPLAY	O.; O.INTEGRITY; O.LABELS; O.SECURE_TRANS; O.VALIDATE
T.UNAUTH_ACCESS	O.ACCESS_CTRL; O.I&A; O.ISOLATE; O.LIMIT_RESTRICTION; O.USER_ATTRIBUTES; O.USER_AUTH_MGMT; OIE.CRYPTO_OPERATION
P.ACCOUNTABILITY	O.ACCESS_CTRL;O.I&A; O.SECURITY_ROLES; O.USER_AUTH_MGMT
P.AVAILABILITY	O.ACCESS_CTRL; O.I&A; O.LIMIT_RESTRICTION; O.SECURITY_ROLES; O.USER_ATTRIBUTES; O.USER_AUTH_MGMT; OIE.CRYPTO_OPERATION

Table 6.1. Objective to Policy/Threat/Assumption Mapping

Threats/Policies	Objectives
P.GUIDANCE	O.INSTALL; O.TRAIN
P.INFORMATION_AC	O.ACCESS_CTRL; O.I&A; O.LIMIT_RESTRICTION; O.USER_AUTH_MGMT
P.INTEGRITY	O.INTEGRITY; O.ISOLATE; O.VALIDATE ;OIE.CRYPTO_OPERATION
P. PHYSICAL_ACCESS	O. ACCESS_CTRL; O.USER_AUTH_MGMT;
P. TRAIN	O.GUIDANCE; O.SECURITY_ROLES; O.USER_AUTH_MGMT

Table 6.1.(Cot'd) Objective to Policy/Threat/Assumption Mapping

6.2 Threats Rationale

T.ADMIN_ERROR The browser administrator performs actions that result in an unauthorized browser user having access to browser information.
A browser administrator commits errors or fails to perform essential function and that may directly compromise organizational security objectives or changes the technical security policy enforced by the system or application.
In General, T.ADMIN_ERROR is addressed by:

1. **O.I&A**: This objective counters this threat by requiring the TOE administrator to be authenticated in order to perform their functions.

2. **O.ISOLATE**: This objective counters this threat by requiring web user information to be separated and protected from incidental modifications. Intended modifications to one specific web user attributes will not affect the attributes of others.

3. **O.SECURITY_ROLES**: This objective counters this threat by specifying that only the TOE administrator will be capable of accessing restricted configuration/content of the TOE based on their associated role after authentication.

T.EXPLOIT_ACTIVE_CONTENT Web browser's active content performs operations on local platform that are undesired or unknown by browser user.
This is a broad category that includes most well known browser vulnerabilities related to the use of Java, Javascript, and ActiveX. The fundamental risk is a violation of user assumptions about the trust placed in active content and the ability of the browser to enforce expected constraints on the active content (e.g. a "sandbox" model, code signing, history-based access control, proff carrying code). Denials of service attacks are a special subset of this threat category that is common in well-known exploits (e.g., crashing the system, whiting out the screen).
In General, T.EXPLOIT_ACTIVE_CONTENT is addressed by:

1. **O.ACTIVE_CONTENT**: Active content downloaded by the web browser is controlled. This objective requires that the browser be capable of limiting the ability of active content

(e.g. mobile code written in Java) to perform potentially harmful actions by limiting access to system and other user programs and data.

T.EXPLOIT_USER_ACCESS A hostile web application can escape the browser containment to access operating system files.

The security model for active content assumes that the browser provides isolation between operating system files and the TOE. This containment should preclude the TOE from accessing the platform system files. If an application is able to breach this containment, it can use the browser user's access rights to perform operations that are not permitted by the remote user.
In General, T.EXPLOIT_USER_ACCESS is addressed by:

1. **O.ACCESS_CTRL**: This objective counters this threat by limiting accessibility and visibility of the TOE only to those users specifically authorized by authentication mechanisms, to include the host platform authentication. Users not identified shall be strictly prohibited accessing TOE information.

2. **OIE.CRYPTO_SERVICES**: This objective counters this threat by invoking the use of various cryptographic services for user authentication. This authentication process must provide non-repudiated evidence of the user's identity and access authorizations.

T.HACKER_ACCESS A hacker gains access due to the TOE allowing an unauthorized browser user to view browser information

The hacker may have the capability to view browser data by attacking the client during the authentication process, injecting viruses, or infecting the TOE with malicious code.
In General, T.HACKER_ACCESS is addressed by:

1. **O.DATA_EXCH_CONF**: This objective, in conjunction with OIE.CRYPRO_OPERATION, counters this threat in order to encrypt TOE data to prevent unauthorized disclosure.

2. **O.DATA_EXPOSURE**: This objective, in conjunction with OIE.CRYPRO_OPERATION, counters this threat in order to encrypt TOE data to prevent unauthorized disclosure.

3. **O.I&A**: This objective counters this threat by requiring each TOE user to be authenticated in order to perform any function or view information attributed to them.

4. **O.INTEGRITY**: This objective counters this threat by providing a reliable assurance that the TOE data originates from its intended source, and is protected through secure channel transmission.

5. **O.SESSION_TERMINATION**: This objective counters this threat by establishing activity timeout intervals on established sessions. Thus, hackers attempting to access resources may cease their malicious activities in order to remain undetected by security controls.

6. **OIE.CRYPTO_SERVICES**: This objective counters this threat by invoking the use of various cryptographic services for user authentication. This authentication process must provide non-repudiated evidence of the user's identity and access authorizations.

T.MALICIOUS_CODE Authorized browser user or hacker stores or executes malicious code causing the TOE to function improperly.
The unauthorized and/or authorize user or attacker causes abnormal processing to occur that will violate the integrity, availability, or confidentiality of the TOE.
In General, T.MALICIOUS_CODE is addressed by:

1. **O.ACCESS_CTRL**: This objective counters this threat by limiting accessibility and visibility of the TOE only to those users specifically authorized by authentication mechanisms, to include the host platform authentication. After authentication, users will only be granted access to specifics of the TOE based upon their security roles.

2. **O.I&A**: This objective counters this threat by requiring the TOE user to be authenticated in order to perform their functions and view information attributed to them. Those individuals not authenticated will be strictly denied access to all levels of the TOE.

3. **O.INTEGRITY**: This objective counters this threat by providing a reliable assurance that the TOE data originates from its intended source, and is protected through secure channel transmission.

T.MOD_BROWSER_CONFIG Unauthorized users modify browser security policy.

Potentially damaging code (e.g., viruses) could modify the files that reflect permissions for subsequent operations, or a rogue Java applet (if given permission erroneously) could modify policy files or security configuration preferences of an authorized user.
In General, T.MOD_BROWSER_CONFIG is addressed by:

1. **O.ACCESS_CTRL**: This objective counters this threat by limiting accessibility and visibility of the TOE only to those users specifically authorized by authentication mechanisms coinciding with host platform. Users not identified by either shall be strictly prohibited further actions.

2. **O.I&A**: This objective counters this threat by restricting the ability to modify a TOE configuration only to those individuals properly identified and authenticated as the TOE administrator.

3. **O.USER_AUTH_MGMT**: This objective supports counter this threat by requiring the TOE administrator to accurately manage user privileges in accordance with their security roles and policies. Accurate management serves to prohibit inadvertent or unauthorized modification of TOE configuration..

T.RESIDUAL_DATA Browser user views residual data that had been written by a different browser user.
A browser user finds information that was left in the system by another browser user. Any time that a browser stores potentially sensitive information on disk (e.g., temporary files, cached application data, message buffers), it must be protected from access to other users. If private information is made available to other browser users, this can lead to unintended disclosure of user information. Once this flaw is discovered, it can be exploited to gather belonging to the system or another user.
In General, T.RESIDUAL_DATA is addressed by:

1. **O.ISOLATE**: This objective counters this threat by requiring web user information to be separated and protected from viewing or modifications by other TOE users. Information written to the TOE (e.g., cache) will be attributed only to the user of origin and inaccessible by all others.

T.SESSION_TERMINATE An unattended active session of the TOE may results in an unauthorized user obtaining browser information or performing unauthorized functions.
The risk is that an authorized user could leave an established secured session unattended. This could allow an unauthorized user and/or attacker to access to browser information. Additionally, the unauthorized user may perform intentional or unintentional malicious acts under the identity of the browser user.
In General, T.SESSION_TERMINATE is addressed by:

1. **O.AUTOMATIC_FUNCTIONS**: This objective counters this threat by enforcing the TOE to automatically invoke a secure state by prohibiting further actions of web users if a security function does not complete successfully in the presence of certain types of failures (e.g., certificate validation failures, invalid authentication credentials.)

2. **O.SESSION_TERMINATION**: This objective counters this threat by closing all active sessions immediately in the event of a session disrupt. This will eliminate virtual active connections by all TOE users regardless of role, and require the TOE user to re-authenticate in order to re-establish a TOE session.

T.SECURE_DISPLAY Deliberate or inadvertent misrepresentation of security-relevant system mode causes browser user to unknowingly violate security policy.
For example, TOE often display the security status of the current link (e.g., a closed padlock to indicate an SSL session). The knowledge of this status, or mode, tells the user if certain actions are permitted or not (e.g., sending passwords in the cleartext vs. encrypted transmissions) A display indicating SSL is active when in fact it is not could result in sensitive data being sent in the clear.
In General, T.SECURE_DISPLAY is addressed by:

1. **O.INTEGRITY**: This objective counters this threat by providing a reliable assurance that the TOE data originates from its intended source, and is protected through secure channel transmission.

2. **O.LABELS**: This objective counters this threat by requiring the appropriate labels be applied to all content displayed by the TOE as it originates from the destination server.

3. **O.SECURE_TRANS**: This objective counters this threat by employing the use of secure protocols that will circumvent the disclosure or modification of information displayed to the TOE user from the destination server.

T.UNAUTH_ACCESS Browser information (e.g., cookies, cache, history list) is read or modified by an unauthorized browser user.

This is essentially a violation of browser-related discretionary access control. Data associated with one user (e.g., cookie list or cache) is read or modified by an unauthorized user. This is particularly important for browsers used in public devices such as web kiosks or public telephones, or shared PCs (e.g., laptops, email facilities in airports, etc.).

In General, T.UNAUTH_ACCESS is addressed by:

1. **O.ACCESS_CTRL**: This objective counters this threat by limiting accessibility and availability of the TOE to only those users specifically authorized by authentication mechanisms coinciding with the host platform. Users not identified shall be strictly prohibited from accessing the TOE.

2. **O.I&A**: This objective counters this threat by requiring the TOE administrator and users be authenticated in order to perform functions on the TOE according to their security roles.

3. **O.ISOLATE**: This objective counters this threat by requiring web user information to be separated and protected from incidental modifications. Information viewed and stored by a specific web user is not accessible by other TOE users.

4. **O.LIMIT_RESTRICTION**: This objective counters this threat by restricting modification of any TOE characteristics prior to proper validation through identification and authentication mechanisms. The restriction would prohibit the deliberate or accidental modification of security mechanisms implemented within the TOE.

5. **O.USER_ATTRIBUTES**: This objective counters this threat by allowing users to access TOE data after certificate-based authentication, that coincides with the individual attributes defined for TOE data paths.

6. **O.USER_AUTH_MGMT**: This objective supports counter this threat by requiring the TOE administrator to accurately manage user privileges in accordance with their security roles and policies. Accurate management serves to prohibit inadvertent or unauthorized disclosure of separate TOE user information.

7. **OIE.CRYPTO_OPERATION**: This objective supports this policy by requiring that TOE access be available through sessions established via Secure Socket Layer (SSL) version 3 channels according to mutual algorithms between the TOE and authorized web users.

Additionally, access to TOE data will be granted via Access Control Lists (ACLs) and certificate-based user authentication.

6.3 Policies Rationale

P.Accountability Individual accountability
Individuals shall be held accountable for their actions.
In General, P.Accountability is addressed by:

1. **O. ACCESS_CTRL**: This objective supports this policy by limiting accessibility of the TOE components and content only to those users specifically authorized and held accountable for their actions through access control mechanisms.

2. **O.I&A**: This objective supports this policy by requiring all individuals (users and administrators) to be authenticated prior to attempting to access the TOE.

3. **O.SECURITY_ROLES**: This objective counters this threat by prohibiting TOE users, not identified by authentication mechanisms as being TOE administrators, from modifying TOE properties based on their role capabilities associated with their access levels.

4. **O.USER_AUTH_MGMT**: This objective supports this policy by requiring the TOE administrator to accurately manage user privileges in accordance with their security roles and policies. Accurate management serves to prohibit inadvertent or unauthorized disclosure of TOE data.

P.Availability Information availability
Information shall be available to satisfy mission requirements.
In General, P.Availability is addressed by:

1. **O. ACCESS_CTRL**: This objective supports this policy by ensuring the availability of the TOE functions to authorized users validated by authentication mechanisms based upon their security roles.

2. **O.I&A**: This objective supports this policy by requiring all individuals to present certificate-based authentication credentials for non-repudiated identification, in order to access TOE functionality available to them based upon security roles.

3. **O.LIMIT_RESTRICTION**: This objective supports this policy by prohibiting all access to the TOE functions prior to user authentication.

4. **O.SECURITY_ROLES**: This objective supports this policy by requiring that the individual's certificate-based authentication and associated access roles be defined prior to allowing the individual (user(s) and/or administrator(s)) access to the TOE.

5. **O.USER_ATTRIBUTES**: This objective supports this policy by making TOE data available to users after authentication (i.e., authentication via certificates and security roles). User attributes are compared to security roles for which access rules to TOE data has been placed upon.

.

P.Guidance Installation and usage guidance
Guidance shall be provided for the secure installation and use of the system.
In General, P.Guidance is addressed by:

1. **O. INSTALL**: This object supports this policy by requiring that the TOE be installed and configured according to the guidance provided to the TOE administrators, and in such a manner to maintain system security.

2. **O.GUIDANCE**: This object supports this policy by requiring that all TOE administrators be given proper guidance and training in the establishment and maintenance of sound security practices for the TOE

P.Information_AC Information access control.
Information shall be accessed only by authorized individuals and processes.
In General, P.Information_AC is addressed by:

1. **O.ACCESS_CTRL**: This objective supports this policy by ensuring the availability of the TOE functions to authorized users validated by authentication mechanisms based upon their security roles.

2. **O.I&A**: This objective supports this policy by requiring all individuals to present certificate-based authentication credentials in order to access TOE functionality available to them based upon security roles and user attributes.

3. **O.LIMIT_RESTRICTION**: This objective supports this policy by prohibiting all access to the TOE functions prior to user authentication.

4. **O.USER_AUTH_MGMT**: This objective supports this policy by requiring the TOE administrator to accurately manage user privileges in accordance with their security roles and policies. Accurate management serves to prohibit inadvertent or unauthorized disclosure of TOE data.

.

P.Integrity Information content integrity.
Information shall retain its content integrity.
In General, P.Integrity is addressed by:

1. **O.INTEGRITY**: This objective supports this policy by providing a reliable assurance that the TOE data is protected from unauthorized modification and that the data source is from the designated TOE origin.

2. **O.ISOLATE**: This objective supports this policy by requiring web users' information to be separated and protected in order to prohibit unauthorized viewing or use by individuals other than the user of origin.

3. **O.VALIDATE**: This objective supports this policy by requiring the TOE positively identify through certificate-based validation, the origin source (e.g., hostname address, certificate authority) of the destination server.

4. **OIE.CRYPTO_OPERATION** . This objective supports this policy by requiring that TOE sessions be established via Secure Socket Layer (SSL) version 3 channels according to mutual algorithms between the TOE and authorized web users in order to prevent unauthorized modification

P.Physical_Control: Physical protection
Appropriate authorities shall be immediately notified of any threats or vulnerabilities impacting systems that process their data.
In General, P.Physical_Control is addressed by:

1. **O.ACCESS_CTRL**: This objective supports this policy by limiting accessibility and visibility of the TOE components and content only to those users specifically authorized by authentication mechanisms coinciding with defined security roles.

2. **O.USER_AUTH_MGMT**: This objective supports this policy by requiring the TOE administrator to accurately manage user privileges in accordance with their security roles and policies. Accurate management serves to prohibit inadvertent or unauthorized disclosure of TOE data.

P.Train TOE Training
Information shall be available to satisfy mission requirements.
In General, P.Train is addressed by:

1. **O.GUIDANCE**: This object supports this policy by requiring that all TOE administrators be given proper guidance and training in the establishment and maintenance of sound security practices for the TOE.

2. **O.SECURITY_ROLES**: This objective supports this policy by requiring that all individuals be trained on the specific functionality of the TOE based upon their authentication credentials and defined security roles (user(s) and/or administrator(s)).

3. **O.USER_AUTH_MGMT**: This objective supports this policy by requiring the TOE administrator to accurately manage user privileges in accordance with their security roles and policies. Accurate management serves to prohibit inadvertent or unauthorized disclosure of TOE data.

6.4 Security Requirements Rationale

This section demonstrates that the identified security requirements are suitable to meet the security objectives contained in this PP.

6.4.1 Functional Security Requirements Rationale

Table 6.2 summaries how each functional requirement, as well as the underlying requirement for a CAPP-compliant hot platform, serves to address the objective of this profile.

Objectives	Requirements
O.ACCESS_CTL	FDP_ACC.2, FDP_ACF.1; FIA_AFL.1,FIA_ATD.1, FIA_SOS.2, FIA_UAU.2, FIA_UAU.3, FIA_UAU.5, FIA_UID.2, FMT_MOF.1, FMT_MSA.1; FMT_MSA.3, FMT_SMR.1, FTA_LSA.1, FTA_TSE.1
O.ACTIVE_CONTENT	FTP_ITC.1, FTP_TRP.1
O.AUTOMATIC_FUNCTIONS	FPT_AMT.1; FPT_FLS.1; FPT_RCV.3; FPT_TST.1
O.DATA_EXPOSURE	FDP_UCT.1; FDP_UIT.1
O.DATA_EXCH_CONF	FCS_COP.1; FDP_ACC.2; FDP_ETC.2; FDP_UCT.1; FMT_MOF.1; FTA_TSE.1
O.I&A	FIA_AFL.1; FIA_SOS.2; FIA_UAU.2; FIA_UAU.3; FIA_UAU.5; FIA_UID.2; FMT_MOF.1; FMT_MSA.1; FPT_STM.1; FTA_TSE.1
O.INTEGRITY	FDP_ACC.2; FDP_DAU.2; FDP_ETC.2; FPT_ITI.1; FPT_TST.1
O.ISOLATE	FDP_RIP.2; FPT_RCV.3; FPT_SEP.1
O.LABELS	FDP_IFC.2; FDP_IFF.1
O.LIMIT_RESTRICTION	FIA_UAU.2; FIA_UAU.3; FIA_UAU.5; FIA_UID.2; FMT_MSA.1; FTA_LSA.1
O.SECURE_TRANS	FCS_COP.1; FDP_ACC.2; FDP_ETC.2; FDP_UCT.1; FTA_TSE.1; FTP_TRP.1
O.SECURITY_ROLES	FMT_MOF.1; FMT_MSA.1; FMT_SMR.1
O.SESSION_TERMINATION	FMT_SAE.1; FTA_SSL.3
O.USER_ATTRIBUTES	FDP_ACC.2; FMT_MSA.1; FMT_MSA.3; FMT_SMR.1

Table 6.2 Functional Component to Security Objective Mapping

Objectives	Requirements
O.USER_AUTH_MGMT	FDP_ACF.1; FIA_ATD.1; FMT_MSA.1; FMT_MSA.3; FMT_SMR.1
O.VALIDATE	FDP_DAU.2

Table 6.2(Cot'd) Functional Component to Security Objective Mapping

O.ACCESS_CTL The host platform must allow the user to determine access controls of data and resources within that user's domain.
The user may want to further restrict execution of active content to protect sensitive areas of information within that user's domain.
O.ACCESS_CTL is implemented in the TOE by:

1. **FDP_ACC.2: Complete access control**
   This component provides the initial definition of the subjects and objects of the access control SFP, whose goal is to provide restrictions on the access of content to authorized web users.

2. **FDP_ACF.1: Security attribute based access control**
   This component defines the rules of access for the web users to content based upon security attributes of the host platform and TOE.

3. **FIA_AFL.1: Authentication failure handling**
   This component defines the rules for defining the number of unsuccessful authentication attempts, as well as the TOE administrator notification and user lockout actions to occur when the threshold has been reached.

4. **FIA_ATD.1: User attribute definition**
   This component defines the rules for the specification of security attributes belonging to individuals that is used by the TSF.

5. **FIA_SOS.2: Specification of secrets**
   This component defines the rules to enforce the generation of secrets (e.g., presentation of credentials) used during the process of authenticating & validating web users.

6. **FIA_UAU.2: User authentication before any action**
   This component requires that the web user successfully authenticated before any attempting TSF actions.

7. **FIA_UAU.3: Unforgeable authentication**
   The component requires that the TSF detect the use of authenticated data that has been forged by any user. The component also requires that TSF prevent authentication data that has been copied from any user.

8. **FIA_UAU.5: Multiple authentication mechanisms**
   This component specifies that TOE users must present not only a userid/password, but a digital certificate for authentication which must be compared to stored TSF user information.

9. **FIA_UID.2: User identification before any action**
   This component requires the TSF to prohibit any further actions on the TOE until the user has been fully authenticated.

10. **FMT_MOF.1: Management of security functions Behavior**
    This component serves to restrict the ability to manipulate the security functions to the TOE administrator.

11. **FMT_MSA.1: Management of security attributes**
    This component serves to reserve the privilege of modifying TOE user security attributes only to the TOE administrator.

12. **FMT_MSA.3: Static attributes initialization**
    This component requires that the TSF provide default values for security attributes to be used. It component also specifies the roles that are allowed to change the default values of the security attributes.

13. **FMT_SMR.1: Security roles**
    This component defines the separate roles between the TOE administrator and user.

14. **FTA_LSA.1: Limitation on scope of selectable attributes**
    This component defines the rules for determining which session security attributes to use based upon the authentication credentials presented to the TOE.

15. **FTA_TSE.1: TOE session establishment**
    This component defines the rules for denying session establishment using the TOE to a destination server.

O.ACTIVE_CONTENT Active content downloaded from a Web Server is controlled.
This objective requires that the browser be capable of limiting the ability of active content (e.g. mobile code written in Java) to perform potentially harmful actions by limiting access to system and other user programs and data.
O.ACTIVE_CONTENT is implemented in the TOE by:

1. **FPT_ITC.1**: Inter-TSF trusted channel
   This component defines the rules for protecting active content data from unauthorized disclosure during transmission between the TSF and a remote trusted IT product

2. **FTP_TRP.1**: Trusted path

This component defines the rules for the TSF to establish of a logically distinct communication path between the TOE and destination web servers that provides assured identification and protection of communication data.

O.AUTOMATIC_FUNCTIONS Browser recovers automatically to a consistent, secure state if a TSF does not complete successfully in the presence of certain types of failures..
The TOE will return to a static page display if failures of the TSF occur.
O.AUTHENTICATE is implemented in the TOE by:

1. **FPT_AMT.1**: Abstract machine testing
   This component defines the rules for the TSF to run a series of consistency tests during initial start-up and periodically during normal operation of the TOE to demonstrate correct operation of the security features.

2. **FPT_FLS.1**: Fail with preservation of secure state
   This component lists the types of failures in the TSF for which the TSF should fail to a secure state. These failures include, but are not limited to, session interruptions, hardware failures, and operational environment failures.

3. **FPT_RCV.3**: Automated recovery without undue loss
   This component provides for the automated recovery of the TOE to a secure state in the event of a web session failure or services disruption

4. **FPT_TST.1**: TSF testing
   This component defines the types of self-tests to be run during initial start-up and periodically to demonstrate the correct operation of the TSF. Additionally, this component defines the roles of TOE users that are authorized to verify the integrity of TSF data and executable code.

O.DATA_EXPOSURE User information that is provided by the TOE to remote servers must be adequately protected from exposure.
Users can be deceived into violating security policy if they are not trained to understand the ways in which information can be disclosed when it is transmitted to a web server.
O.DATA_EXPOSURE is implemented in the TOE by:

1. **FDP_UCT.1**: Basic data exchange confidentiality
   This component requires the transmission and reception of information is done in a manner that prevents unauthorized disclosure of content.

2. **FDP_UIT.1**: Data exchange integrity
   This component requires the TSF to enforce controls in order to detect the modification of transmitted data, as well as, the ability to verify whether data has been modified upon its receipt.

O.DATA_EXCH_CONF The TSF shall enforce confidentiality of data exchanged between the web server and the TOE

The objective ensures that the data is protected by the use of encryption prior to being transmitted to/from the browser user.

O.DATA_EXCH_CONF is implemented in the TOE by:

1. **FCS_COP.1**: Cryptographic operation
   This component performs a set of cryptographic operations, such as data encryption and/or certificate authentication that is used to protect the content during transmission to and from the TOE.

2. **FDP_ACC.2**: Complete access control
   This component specifies the unique access control SFP and protects the content from disclosure during transmission..

3. **FDP_ETC.2**: Export of user data with security attributes
   This component requires the security attributes of transmitted information be included with the information.

4. **FDP_UCT.1**: Basic data exchange confidentiality
   This component requires the transmission and reception of information is done in a manner that prevents unauthorized disclosure of content.

5. **FMT_MOF.1**: Management of security functions behavior
   This component serves to restrict the ability to manipulate the security functions to the TOE administrator.

6. **FTA_TSE.1**: TOE session establishment
   This component defines the rules for denying session establishment using the TOE to a destination server.

O.I&A The browser user shall be required to provide evidence (e.g., digital certificate) to be positively identified and authenticated to support accountability by the destination servers. This authentication, working in conjunction with server security mechanisms, shall provide for the capabilities of auditing each authenticated use*r.*

This objective establishes the association of each transaction between an authenticated user and an application with a unique transaction ID. This allows events associated with a given transaction to be distinguished from other events involving the user and the application

O.I&A is implemented in the TOE by:

1. **FIA_SOS.2**: Specification of secrets
   This component defines the rules to enforce the generation of secrets (e.g., presentation of credentials) used during the process of authenticating & validating web users.

2. **FIA_UAU.2**: User authentication before any action

This component requires that the web user successfully authenticated before any attempting TSF actions.

3. **FIA_UAU.3**: Unforgeable authentication
   The component requires that the TSF detect the use of authenticated data that has been forged by any user. The component also requires that TSF prevent authentication data that has been copied from any user.

4. **FIA_UAU.5**: Multiple authentication mechanisms
   This component specifies that TOE users must present not only a userid/password, but a digital certificate for authentication which must be compared to stored TSF user information.

5. **FIA_UID.2**: User identification before any action
   This component requires the TSF to prohibit any further actions on the TOE until the user has been fully authenticated.

6. **FMT_MOF.1**: Management of security functions behavior
   This component serves to restrict the ability to manipulate the security functions to the TOE administrator.

7. **FMT_MSA.1**: Management of security attributes
   This component serves to reserve the privilege of modifying TOE user security attributes only to the TOE administrator.

8. **FPT_STM.1**: Reliable time stamps
   This component requires that the TSF provide reliable time stamps. This is critical for the accountability of user actions.

9. **FTA_TSE.1**: TOE session establishment
   This component specifies that the TSF can deny session establishment based on security attribute authentication.

.

O.INTEGRITY Browser users must be able to trust that the received information is from the expected originating source.
The goal of this objective is to provide browser users adequate confidence that the information received came from the expected source. Typically this is done by verification of the certificate held by the web server, which is transmitted as part of the HTTPS protocol.
O.INTEGRITY is implemented in the TOE by:

1. **FDP_ACC.2**: Complete access control
   This component specifies the unique access control SFP and protects the content from disclosure during transmission.

2. **FDP_DAU.2**: Data authentication with identify of guarantor

This component requires that the web users be able to verify the identity of the transmitter of the information.

3. **FDP_ETC.2**: Export of user data with security attributes
   This component requires the security attributes of transmitted information be included with the information.

4. **FPT_ITI.1**: Inter-TSF detection of modification
   This component requires the

5. **FPT_TST.1**: TSF testing
   This component defines the types of self-tests to be run during initial start-up and periodically to demonstrate the correct operation of the TSF. Additionally, this component defines the roles of TOE users that are authorized to verify the integrity of TSF data and executable code

O.ISOLATE Browser user data (e.g., history, profiles, cookies, and cache) must be separate from other user data and not accessible by other users.
Many users may use the same workstation either simultaneously or at separate times to launch web browser sessions. User-specific data that is maintained for these user sessions must remain separate from other user sessions.
O.ISOLATE is implemented in the TOE by:

1. **FPT_RCV.3**: Automated recovery without undue loss
   This component provides for the automated recovery of the TOE to a secure state in the event of a web session failure or services disruption in order to isolate and protect TSF data.

2. **FDP_RIP.2**: full residual information protection
   This component ensures that content of a resource (e.g., TOE cache) is made unavailable upon the deallocation or termination of a TOE session.

3. **FPT_SEP.1**: TSF domain separation
   This component defines the logical separation of host platform resources for simultaneous operation of the TSF to avoid interference and tampering by untrusted objects.

O.LABELS Information shall be visually displayed, labeled or marked according to their content as received from the destination server.
The objective ensures that the TOE content will be displayed, labeled or marked in order to prevent the disclosure of data from secure web servers to unauthorized web users.
O.LABELS is implemented in the TOE by:

1. **FDP_IFC.2**: Complete information flow control
   This component requires that an information flow control policy that places sensitive access control label on web pages that is to be accessed by authorized users.

2. **FDP_IFF.1**: Simple security attributes
   This component requires that sensitive access control labels on web pages identify authorized users based upon the security attributes.

O.LIMIT_RESTRICTION The TSF shall restrict the action of the web user prior to being authenticated.
The objective limits a browser user's actions (i.e., only a log on is allowed) prior to the TOE successfully communicating with a web server in order to verify the identity of the browser user.
O.LIMIT_RESTRICTION is implemented in the TOE by:

1. **FIA_UAU.2**: User authentication before any action
   This component requires that the web user successfully authenticated before any attempting TSF actions.

2. **FIA_UAU.3**: Unforgeable authentication
   The component requires that the TSF detect the use of authenticated data that has been forged by any user. The component also requires that TSF prevent authentication data that has been copied from any user.

3. **FIA_UAU.5**: Multiple authentication mechanisms
   This component specifies that TOE users must present not only a userid/password, but a digital certificate for authentication which must be compared to stored TSF user information.

4. **FIA_UID.2**: User identification before any action
   This component requires the TSF to prohibit any further actions on the TOE until the user has been fully authenticated.

5. **FMT_MSA.1**: Management of security attributes
   This component serves to reserve the privilege of modifying TOE user security attributes only to the TOE administrator.

6. **FTA_LSA.1**: Limitation on scope of selectable attributes
   This component defines the rules for determining which session security attributes to use based upon the authentication credentials presented to the TOE.

O.SECURE_TRANS Content shall be protected during transmission to and from web server.
Secure transmission protocols shall be employed by the TOE (i.e., SSL, TLS) in order to protect information from unauthorized disclosure or modification. While outside the scope of the PP, additional secure transmission countermeasures (e.g., VPNs) may be used by the host platform system and/or physical network components.
O.SECURE_TRANS is implemented in the TOE by:

1. **FCS_COP.1**: Cryptographic operation
   This component performs a set of cryptographic operations, such as data encryption and/or certificate authentication that is used to protect the content during transmission to and from the TOE

2. **FDP_ACC.2**: Complete access control
   This component provides the initial definition of the subjects and objects of the access control SFP, whose goal is to provide restrictions on the access of content to authorized web users.

3. **FDP_ETC.2**: Export of user data with security attributes
   This component requires the security attributes of transmitted information be included with the information.

4. **FDP_UCT.1**: Basic data exchange confidentiality
   This component requires the transmission and reception of information is done in a manner that prevents unauthorized disclosure of content.

5. **FTA_TSE.1**: TOE session establishment
   This component defines the rules for denying session establishment using the TOE to a destination server.

6. **FTP_TRP.1**: Trusted path
   This component defines the rules for the TSF to establish of a logically distinct communication path between the TOE and destination web servers that provides assured identification and protection of communication data.

O.SECURITY_ROLES The TSF shall maintain user privilege role separation.
This objective allows the TOE administrator to maintain security-relevant roles and the association of users with those roles.
O.SECURITY_ROLES is implemented in the TOE by:

1. **FMT_MOF.1**: Management of security functions behavior
   This component serves to restricts the ability to define security roles to the TOE administrator.

2. **FMT_MSA.1**: Management of security attributes
   This component serves to reserve the privilege of modifying TOE user security attributes only to the TOE administrator.

3. **FMT_SMR.1**: Security roles
   This component defines the separate roles between the TOE administrator and user.

O.SESSION_TERMINATION The TOE shall terminate a session for inactivity during a secure established session.
The objective allows the TOE to terminate an inactive secure session after a specified interval of inactivity.
O.SESSION_TERMINATION is implemented in the TOE by:

1. **FMT_SAE.1**: Time-limited authorization
   This component defines an expiration of time in which an authenticated TOE session will be terminated due to inactivity, and request that the TOE user re-authenticate themselves to continue the session.

2. **FTA_SSL.3**: TSF-initiated termination
   The component specifies that the TSF shall initiate an action to terminate an active TOE session upon reaching a security threshold.

O.USER_ATTRIBUTES The TOE shall maintain the user attributes for each active session.
The objective states that the TSF shall maintain security attributes associated with individual users in addition to browser user identity.
O.USER_ATTRIBUTES is implemented in the TOE by:

1. **FDP_ACC.2**: Complete access control
   This component specifies the unique access control SFP and protects the content from disclosure during transmission.

2. **FMT_MSA.1**: Management of security attributes
   This component serves to reserve the privilege of modifying TOE user security attributes only to the TOE administrator.

3. **FMT_MSA.3**: Static attributes initialization
   This component requires that the TSF provide default values for security attributes to be used. It component also specifies the roles that are allowed to change the default values of the security attributes.

4. **FMT_SMR.1**: Security roles
   This component defines the separate roles between the TOE administrator and user.

O.USER_AUTH_MGMT The TSF shall manage user authorization privileges.
The TSF manages and updates user authorization and privilege data in accordance with organizational security and personnel policies.
O.USER_AUTH_MGMT is implemented in the TOE by:

1. **FDP_ACF.1**: Security attribute based access control
   This component defines the rules of access for the web users to content based upon security attributes of the host platform and TOE.

2. **FIA_ATD.1**: User attribute definition
   This component defines the rules for the specification of security attributes belonging to individuals that is used by the TSF.

3. **FMT_MSA.1**: Management of security attributes
   This component serves to reserve the privilege of modifying TOE user security attributes only to the TOE administrator.

4. **FMT_MSA.3**: Static attributes initialization
   This component requires that the TSF provide default values for security attributes to be used. It component also specifies the roles that are allowed to change the default values of the security attributes.

5. **FMT_SMR.1**: Security roles
   This component defines the separate roles between the TOE administrator and user.

O. VALIDATE The TOE is capable of validating the origin source of the connected destination server.
In other words, the browser can, through the use of validation mechanisms (e.g., PKI server certificates) confirm the identity of the server source with which it is communicating.
O.VALIDATE is implemented in the TOE by:

1. **FDP_DAU.2**: Data authentication with identify of Guarantor
   This component requires that the web users be able to verify the identity of the transmitter of the information.

6.4.2 Assurance Security Requirements Rationale

Assurance is that sessions of the web browser shall provide positive Identification & Authentication and non-repudiation of Internet/Intranet sessions. This assurance shall also provide restrictability of individual user files (e.g., preference settings, and cache) by providing access to the owners of such information. Working in conjunction with the connecting web server, the browser shall provide the appropriate encrypted secure links based on common algorithms set between the two applications.

6.5 Security Functional Requirements Grounding in Objectives

This section provides the mapping of the security functional requirements to the objectives identified in this protection profile. Table 6-3 identifies contains the each selected security functional requirement that satisfies the security objectives.

Requirements	Objectives
FDP_ACC.2	O.ACCESS_CTRL; O.DATA_EXCH_CONF; O.INEGRITY; O.SECURE_TRANS; O.USER_ATTRIBUTES
FDP_ACF.1	O.ACCESS_CTRL ; O.USER_AUTH_MGMT
FDP_DAU.2	O.INTEGRITY; O.VALIDATE
FDP_ETC.2	O.DATA_EXCH_CONF; O.INTEGRITY; O.SECURE_TRANS
FDP_ICF.2	O.LABELS
FDP_IFF.1	O.LABELS
FDP_RIP.2	O.ISOLATE
FDP_UCT.1	O.DATA_EXPOSURE; O.DATA_EXCH_CONF; O.SECURE_TRANS
FDP_UIT.1	O.DATA_EXPOSURE
FIA_AFL.1	O.ACCESS_CTRL; OI&A
FIA_ATD.1	O.ACCESS_CTRL; O.USER_AUTH_MGMT
FIA_SOS.2	O.ACCESS_CTRL
FIA_UAU.2	O.ACCESS_CTL, O.I&A; O.LIMIT_RESTRICTION
FIA_UAU.3	O.ACCESS_CTRL; O.I&A;O.LIMIT_RESTRICTION
FIA_UAU.5	O.ACCESS_CTRL, O.I&A; O.LIMIT_RESTRICTION
FIA_UID.2	O.ACCESS_CTRL O.I&A; O.LIMIT_RESTRICTION
FMT_MOF.1	O.ACCESS_CTRL; O.DATA_EXCH_CONF; O.I&A; O.SECURITY_ROLES
FMT_MSA.1	O.ACCESS_CTRL; O.I&A; O.LIMIT_RESTRICTION; O.SECURITY_ROLES; O.USER_ATTRIBUTES; O.USER_AUTH_MGMT
FMT_MSA.13	O.ACCESS_CTRL; O.USER_ATTRIBUTES; O.USER_AUTH_MGMT
FMT_SAE.1	O.SESSION_TERMINATION

Table 6-3. Requirements to Objectives Mapping

Requirements	Objectives
FMT_SMR.1	O.ACCESS_CTRL; O.SECURITY_ROLES; O.USER_ATTRIBUTES; O.USER_AUTH_MGMT
FPT_AMT.1	O.AUTOMATIC_FUNCTIONS
FPT_FLS.1	O.AUTOMATIC_FUNCTIONS
FPT_ITI.1	O.INTEGRITY
FPT_RCV.3	O.AUTOMATIC_FUNCTIONS; O.ISOLATE
FPT_SEP.1	O.ISOLATE
FPT_STM.1	O.I&A
FPT_TST.1	O.AUTOMATIC_FUNCTIONS; O.INTEGRITY
FTA_LSA.1	O.ACCESS_CTRL ; O.LIMIT_RESTRICTION
FTA_SSL.3	O.SESSION_TERMINATION
FTA_TSE.1	O.ACCESS_CTRL ; O.DATA_EXCH_CONF; O.I&A; O.SECURE_TRANS
FTP_ITC.1	O.ACTIVE_CONTENT
FTP_TRP.1	O.ACTIVE_CONTENT, O.SECURE_TRANS

Table 6-3.(Cot'd) Requirements to Objectives Mapping

6.6 Dependency Rationale

6.6.1 Dependency Requirements Rationale

Table 6-6-1 identifies the dependency allocated to the functional and assurance requirements addressed within this protection profile. Table 6-4 identifies unsupported dependencies and an explanation for their omission from table 6-5.

Requirement	Dependencies
Functional Requirements	
FCS_COP.1	FDP_ITC.1
FDP_ACC.2	FDP_ACF.1
FDP_ACF.1	FMT_MSA.3
FDP_DAU.2	FDP_ACC.1,
FDP_ETC.2	FDP_IFF.1
FDP_IFC.2	FDP_IFF.1
FDP_IFF.1	FMT_MSA.3
FDP_UCT.1	FTP_ITC.1
FDP_UIT.1	FTP_ITC.1
FMT_MOF.1	FMT_SMR.1

Table 6-4 Functional and Assurance Requirements Dependencies

Requirement	Dependencies
Functional Requirements	
FMT_MSA.1	FMT_SMR.1
FMT_MSA.3	FMT_MSA.1; FMT_SMR.1
FMT_SAE.1	FMT_SMR.1; FPT_STM.1
FPT_RCV.3	FPT_TST.1; AGD_ADM.1
FPT_TST.1	FPT_AMT.1
Assurance Requirements	
ADO_IGS.1	AGD_ADM.1
ADV_FSP.1	ADV_RCR.1
ADV_HLD.1	ADV_FSP.1, ADV_RCR.1
AGD_ADM.1	ADV_FSP.1
AGD_USR.1	ADV_FSP.1
ATE_COV.1	ADV_FSP.1, ATE_FUN.1
ATE_IND.2	ADV_FSP.1, AGD_ADM.1, AGD_USR.1, ATE_FUN.1
AVA_SOF.1	ADV_FSP.1, ADV_HLD.1
AVA_VLA.1	ADV_FSP.1, ADV_HLD.1, AGD_ADM.1, AGD_USR.1

Table 6-4 (Cot'd) Functional and Assurance Requirements Dependencies

Unsupported Dependencies	Rationale
Functional Requirements	
FCS_COP.1	FMT_MSA.2 was not selected since initialization attributes for default values could be either restrictive (secure) or permissive (clear).
FDP_ACF.1	FDP_ACC.1 is a hierarchical component of FDP_ACC.2 and by rule the elements are included with the selection of FDP_ACC.2.
FDP_DAU.2	FIA_UAU.1 is an hierarchical component of FIA_UAU.2 and by rule the elements are included with the selection of FIA_UAU.2
FDP_ETC.2	FDP_ACC.1 is an hierarchical component of FDP_ACC.2 and FDP_IFC.1 is an hierarchical component of FDP_IFC.2 so by rule, the elements are included with each respective higher selection.
FDP_IFF.1	FDP_IFC.1 is an hierarchical component of FDP_IFC.2 so by rule, the elements are included with the higher selection of FDP_IFC.2.
FDP_UCT.1	FDP_ACC.1 is an hierarchical component of FDP_ACC.2 and FDP_IFC.1 is an hierarchical component of FDP_IFC.2 so by rule, the elements are included with each respective higher selection.
FDP_UIT.1	FDP_ACC.1 is an hierarchical component of FDP_ACC.2 and FDP_IFC.1 is an hierarchical component of FDP_IFC.2 so by rule,

Unsupported Dependencies	Rationale
Functional Requirements	
	the elements are included with each respective higher selection.
FIA_AFL.1	FIA_UAU.1 is an hierarchical component of FIA_UAU.2 so by rule, the elements are included with the higher selection of FIA_UAU.2.
FIA_UID.2	FIA_UID.1 is an hierarchical component of FIA_UID.2 so by rule, the elements are included with the higher selection of FIA_UID.2.
FMT_MSA.1	FDP_ACC.1 is an hierarchical component of FDP_ACC.2 and FDP_IFC.1 is an hierarchical component of FDP_IFC.2 so by rule, the elements are included with each respective higher selection.
FMT_SMR.1	FIA_UID.1 is an hierarchical component of FIA_UID.2 so by rule, the elements are included with the higher selection of FIA_UID.2.
FPT_FLS.1	ADV_SPM.1 was not selected since the intended TOE should be adapted to meet EAL2 and ADV_SPM.1 is an inclusion of EAL4.
FPT_RCV.3	ADV_SPM.1 was not selected since the intended TOE should be adapted to meet EAL2 and ADV_SPM.1 is an inclusion of EAL4.

Table 6.5. Unsupported Dependencies

Appendix A Acronyms

CAPP	Controlled Access Protection Profile
CC	Common Criteria
CGI	Common Gateway Interface
EAL	Evaluation Assurance Level
FTP	File Transfer Protocol
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
HTTPS	HTTP with a Secure Socket Layer (SSL)
ISSOs	Information System Security Officers
IT	Information Technology
MIME	Multipurpose Internet Mail Extensions
N/A	Not Applicable
PDF	Portable Data File
PP	Protection Profile
SBU	Sensitive But Unclassified
SF	Security Function
SFP	Security Function Policy
SSL	Secure Sockets Layer
SOF	Strength of Function
TLS	Transport Layer Security
TOE	Target of Evaluation
TSC	TSF Scope of Control
TSF	TOE Security Functions

TSP	TOE Security Policy
URL	Universal Resource Locator
WWW	World Wide Web

References

Common Criteria for Information Technology Security Evaluation; Part 2: Security Functional Requirements, CCIB-98-027, August 1999

Common Criteria for Information Technology Security Evaluation; Part 3: Security Assurance Requirements, CCIB-98-028, August 1999

Web Server Protection Profile, Draft Version.0.5, January 26, 2001